



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/893,550	06/29/2001	Ravi Kanth V. Kothuri	19111.0035	4447

23517 7590 01/12/2004

SWIDLER BERLIN SHEREFF FRIEDMAN, LLP  
3000 K STREET, NW  
BOX 1P  
WASHINGTON, DC 20007

EXAMINER

GODDARD, BRIAN D

ART UNIT	PAPER NUMBER
----------	--------------

2171

DATE MAILED: 01/12/2004

2/

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/893,550

Applicant(s)

KOTHURI ET AL.

Examiner

Brian Goddard

Art Unit

2171

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 30 October 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-31 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-31 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 29 June 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
- a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 3.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

## **DETAILED ACTION**

### ***Claim Objections***

1. Claims 7, 11, 13, 30 and 31 are objected to because of the following informalities:

The word 'if' should be inserted between "determining" and "a number of" in the tenth line of claim 7 (or another such appropriate change) because the limitation is clearly directed to determining if a number corresponds to a target number.

The same modification should be made to claims 11 (tenth line), 30 (fourteenth line) and 31 (fourteenth line) because the informality of claim 7 is repeated in these claims.

The word "of" should be removed from the phrase "a memory capacity of in which" in the second line of claim 13 because it is extraneous.

Appropriate correction is required.

### ***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claim 10 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 10 recites the limitation "the distance" in the first line of the claim. There is insufficient antecedent basis for this limitation in the claim.

In the interest of compact prosecution, the examiner assumes that claim 10 should depend from claim 8 instead of claim 7, because claim 8 contains the appropriate antecedent basis for this limitation.

### ***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-3, 5-11, 14, 16, 19-27 and 30-31 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent Application Publication No. 2002/0147703 to Yu et al.

Referring to claim 1, Yu discloses a method for locating neighbor objects of a query geometry in a database as claimed. See Figures 1-5 and the corresponding portions of Yu's specification for this disclosure. Yu teaches a method [See Fig. 5] for locating neighbor objects of a query geometry in a database [50], the method comprising:

1) tiling [partitioning (See ¶ 0007 & 0034 as well as Figs. 6-10)] a query geometry at a tiling level  $t$  [partitioning levels differentiated in Figs. 6-10];

2) conducting a search [initial call to  $\text{SearchO}(q, r)$  function in while loop of  $\text{iDistanceKNN}$  function] of a search area [within initial search radius  $r = \Delta r$ ] comprising interior and boundary tiles [inclusively (See ¶ 0053)] of the query geometry for neighbors [nearest neighbors], if the search does not produce a result set that includes a target number  $[K]$  of neighbors [while loop (while  $|S| < K$ )], then the following steps are carried out starting with a radius  $r$  of 1:

3) expanding [ $r = r + \Delta r$ ] the search area to a radius  $r$  tiles at a distance  $d$  [ $\Delta r$ ] from the boundary tiles and conducting a search [next call to  $\text{SearchO}(q, r)$  function in while loop of  $\text{iDistanceKNN}$  function] for neighbors in the expanded search area, where  $(r-1)*tw < d \leq r*tw$ , wherein  $tw$  is the maximum or minimum of the x-width, y-width of a tile at level  $t$  and  $r$  is the query radius;

4) using each tile in the expanded search area it is determined if there are any geometries at distance  $\leq r*tw$  that are to be included in the result set [call to  $\text{SearchInward}$  function];

5) if a target number of neighbors is not found [if  $|S| < K$ ], steps 3-4 are repeated with a query radius of 2 tile widths followed by 3 tile widths [while loop continues];

6) if the target number still is not reached [if  $O_i$  has been searched before and  $\text{sphere}(O_i, \text{dist\_max}_i)$  intersects  $\text{sphere}(q, r)$ ] retiling the query geometry at a higher level of tiling [shifted to new partitioning ( $lp[i]$  &  $rp[i]$  are shifted)]; and

7) repeating steps 1-7 [while loop continues] until either the target number [K] is reached [K nearest neighbors are found] or all objects in the database are searched [ $r \geq \text{max\_r}$ ].

Referring to claim 2, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 1, as above, "wherein geometries at distances  $> r \cdot \text{tw}$  [if  $O_i$  has not been searched before and  $\text{sphere}(O_i, \text{dist\_max}_i)$  intersects  $\text{sphere}(q, r)$ ] are included in the search with query radius  $r$ " as claimed.

Referring to claim 3, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 1, as above, "wherein only a portion of tiles [those falling completely within  $\text{dist\_max}_i$  of  $q$  (the query object)] in the expanded query radius are materialized prior to searching [See first half of SearchO function] the expanded query radius for neighbors, and wherein additional tiles are materialized [those further from  $q$  than  $\text{dist\_max}_i$ , but whose sphere still intersects the query sphere (See second half of SearchO function)] until all tiles at a specific query radius have been generated, searched and included in a result" as claimed.

Referring to claim 5, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method according to claim 3, as above, "further comprising removing duplicate neighbors [if  $O_i$  is searched a second time it cannot be added to  $S$  (See SearchO function)] from the result  $[S]$ " as claimed.

Referring to claim 6, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method according to claim 3, as above, "further comprising: retaining only the target number  $[K]$  of neighbors  $[S]$  if more than the target number of neighbors is found [See SearchInward function] and terminating the searching process [while loop exits if  $|S| \geq K$ ]" as claimed.

Referring to claim 7, Yu discloses a method for locating neighbor objects of a query object in a database as claimed. See Figures 1-5 and the corresponding portions of Yu's specification for this disclosure. Yu teaches a method for locating neighbor objects of a query object in a database, the method comprising:

defining a query tile set [partitioning (See ¶ 0007 & 0034 as well as Figs. 6-10)] comprising a plurality of tiles [partitions] that define an initial query radius [querydist =  $\Delta r$ ] that the query object is grouped within [See ¶ 0057];

locating neighbor objects within the initial query radius [initial call to SearchO(q, r) function in while loop of iDistanceKNN function];

determining a number  $[|S|]$  of neighbor objects  $[S]$  within the initial query radius;

expanding the query radius  $[r = r + \Delta r]$  beyond the initial query radius by defining an expanded query tile set beyond the query tile set if [if  $|S| < K$ ] the number of neighbor objects is less than a target number  $[K]$ ;

locating neighbor objects within the expanded query radius [next call to SearchO(q, r) function in while loop of iDistanceKNN function];

determining if  $[if \ |S| < K]$  a number of neighbor objects within the expanded radius corresponds to a target number; and

continuing to expand the query radius and locate neighbor objects [while loop continues] within the expanded radius until [while  $|S| < K$ ] the number of neighbor objects equals or exceeds the target number.

Referring to claim 8, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, further comprising: "calculating a distance [ $dis = \text{dist}(O_i, q)$ ] of each neighbor object [ $O_i$ ] from the query object [ $q$ ]" as claimed.

Referring to claim 9, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification, as well as paragraphs 0036-0052, for the details of this disclosure. Yu teaches the method of claim 8, as above, "wherein the distance of each neighbor object [ $O_i$ ] from the center of [the data point] the query object [ $q$ ] is calculated" as claimed.

Referring to claim 10, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification, as well as paragraphs 0036-0052, for the details of this disclosure. Yu teaches the method of claim 8, as above, "wherein the distance is based upon... boundary-to-boundary [point to point] distance" as claimed.



Claim 11 is rejected on the same basis as claim 1, in light of the basis for claim 7 above. See the discussion regarding steps 6 – 7 of claim 1 above in light of the basis for claim 7 for the details of this disclosure.

Referring to claim 14, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, further comprising: "calculating a distance [ $\text{dist}(e_i, q)$ ] of each located neighbor object [ $e_i$ ] from the query object [ $q$ ] if the number [ $|S|$ ] of neighbor objects in the initial query radius exceeds the target number [See first half of SearchInward function] to determine the target number [ $K$ ] of nearest neighbor objects" as claimed.

Referring to claim 16, Yu discloses the method for locating neighbor objects as claimed. See Figures 6-10 and the corresponding portions of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, "wherein all of the tiles [partitions] have the same shape and size [See ¶ 0007 & 0076-0082]" as claimed.

Referring to claim 19, Yu discloses the method for locating neighbor objects as claimed. See Figures 6-10 and the corresponding portions of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, further comprising: "assigning each tile a tile code [reference point] and an associated geometry [cluster shaping] as the tiles [partitions] are defined" as claimed.

Referring to claim 20, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, "wherein the query radius [ $r$ ] is

expanded  $[r = r + \Delta r]$  a selected distance  $[\Delta r]$  and the method further comprises determining a number of tiles [partitions] necessary to expand the query radius the selected distance [See SearchO function]" as claimed.

Referring to claim 21, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, "wherein the query radius  $[r]$  is expanded  $[r = r + \Delta r]$  a selected number of tiles  $[\Delta r = \text{dist\_max}_i]$ " as claimed.

Referring to claim 22, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, "wherein the initial query radius [querydist or  $\Delta r$ ] and the expanded query radius  $[r]$  comprise the same number of tiles" as claimed.

Referring to claims 23 and 24, Yu discloses the method for locating neighbor objects as claimed. See Figures 4-10 and the corresponding portions of Yu's specification for this disclosure. Yu teaches the method according to claim 7, wherein the query object  $[q]$  is grouped completely [is a point residing in] within a plurality of tiles [at least one partition of the initial query radius] as claimed.

Referring to claim 25, Yu discloses the method for locating neighbor objects as claimed. See the Abstract, Background and Summary of the invention for this disclosure. Yu's database [50] comprises a spatial database as claimed.

Referring to claim 26, Yu discloses the method for locating neighbor objects as claimed. See paragraph 0034 for the details of this disclosure. Yu's neighbor objects are organized in order of distance as claimed.

Referring to claim 27, Yu discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Yu's specification for this disclosure. Yu teaches the method of claim 7, as above, further comprising:  
"calculating a distance [ $\text{dist}(e_i, q)$ ] of each located neighbor object [ $e_i$ ] from the query [ $q$ ] to determine the target number [ $K$ ] of nearest neighbor objects [See SearchInward function]" as claimed.

Claim 30 is rejected on the same basis as claim 7. See the discussion regarding claim 7 above for the details of this disclosure. Yu teaches a computer program product [iDistance software] for performing a process for locating neighbor objects of a query object in a database in a computer system [See Fig. 1], comprising:

a computer readable medium [30, 40, 50]; and  
computer program instructions [iDistance function code], recorded on the computer readable medium, executable by a processor [CPU (not numbered)] for performing the steps of...[See claim 7].

Claim 31 is rejected on the same basis as claim 7. See the discussion regarding claims 7 and 30 for the details of this disclosure.

4. Claims 1-31 are rejected under 35 U.S.C. 102(b) as being anticipated by the article entitled "Distance Browsing in Spatial Databases" by Hjaltason et al.

Referring to claim 1, Hjaltason discloses a method for locating neighbor objects of a query geometry in a database as claimed. See Section 4 of Hjaltason's article for this disclosure. Hjaltason teaches a method [enhanced incremental nearest neighbor algorithm (See Fig. 5)] for locating neighbor objects [nearest neighbors] of a query geometry in a database, the method comprising:

1) tiling [spatial database indexed by node regions (See section 4.1)] a query geometry [spatial region] at a tiling level  $t$  [hierarchical level of the tree];

2) conducting a search [initial search of *search region* (See Fig. 2)] of a search area [*search region*] comprising interior and boundary tiles [See Fig. 2] of the query geometry for neighbors, if the search does not produce a result set that includes a target number  $[k]$  of neighbors, then the following steps are carried out starting with a radius  $r$  of 1:

3) expanding the search area to a radius  $r$  tiles at a distance  $d$  from the boundary tiles [See pg. 275] and conducting a search for neighbors in the expanded search area, where  $(r-1)*tw < d \leq r*tw$ , wherein  $tw$  is the maximum or minimum of the x-width, y-width of a tile at level  $t$  and  $r$  is the query radius [See section 4.5];

4) using each tile [node] in the expanded search area it is determined if there are any geometries [objects] at distance  $\leq r*tw$  [within the expanded search region] that are to be included in the result set;

5) if a target number of neighbors is not found, steps 3-4 are repeated with a query radius of 2 tile widths followed by 3 tile widths [while loop continues effectively expanding the search region];

6) if the target number still is not reached retiling the query geometry at a higher level of tiling [jumping to a higher level of recursion in the hierarchy (See 4.1-4.5)]; and

7) repeating steps 1-7 [See Fig. 5] until either the target number [ $k$ ] is reached or all objects in the database are searched [the queue is empty].

Referring to claim 2, Hjaltason discloses the method for locating neighbor objects as claimed. See Figures 3 & 5 and the corresponding portions of Hjaltason's article for this disclosure. Hjaltason teaches the method of claim 1, as above, "wherein geometries [node regions] at distances  $>r*tw$  [non-leaf nodes (See steps 16-20 of Fig. 3 and steps 23-36 of Fig. 5)] are included in the search with query radius  $r$ " as claimed.

Referring to claim 3, Hjaltason discloses the method for locating neighbor objects as claimed. See Figure 5 and the corresponding portion of Hjaltason's article for this disclosure. Hjaltason teaches the method of claim 1, as above, "wherein only a portion [leaf nodes] of tiles [nodes] in the expanded query radius [See sections 4.2-4.5] are materialized prior to searching the expanded query radius for neighbors, and wherein additional tiles [non-leaf nodes] are materialized until all tiles at a specific query radius have been generated, searched and included in a result" as claimed.

Referring to claim 4, Hjaltason discloses the method for locating neighbor objects as claimed. See page 307 for the details of this disclosure. Hjaltason teaches the

method of claim 3, as above, wherein about 1000 to about 5000 [1000, 2000, 4000, ...] tiles are materialized prior to searching as claimed.

Referring to claim 5, Hjaltason discloses the method for locating neighbor objects as claimed. See page 276 for the details of this disclosure. Hjaltason teaches the method of claim 3, as above, further comprising removing duplicate neighbors [See lines 6-8 of Fig. 3] from the result as claimed.

Referring to claim 6, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 4.2-4.5, especially page 281, for the details of this disclosure. Hjaltason teaches the method of claim 3, as above, further comprising: retaining only the target number  $[k]$  of neighbors if more than the target number of neighbors is found [tree traversal is not complete at the particular query radius] and terminating the searching process as claimed.

Referring to claim 7, Hjaltason discloses a method for locating neighbor objects of a query object in a database as claimed. See Section 4 of Hjaltason's article for this disclosure. Hjaltason teaches a method [See Fig. 5] for locating neighbor objects of a query object in a database, the method comprising:

defining a query tile set [grid of node regions (See section 4.1)] comprising a plurality of tiles [nodes] that define an initial query radius [See Fig. 2] that the query object is grouped within [See section 4.2];

locating neighbor objects [nearest neighbor objects] within the initial query radius [See sections 4.2-4.5];

determining a number of neighbor objects [number of times an object (*Element*) is reported] within the initial query radius [in the first pass];

expanding the query radius beyond the initial query radius by defining an expanded query tile set beyond the query tile set [See pg. 275] if the number of neighbor objects is less than a target number [ $k$ ];

locating neighbor objects within the expanded query radius [See sections 4.2-4.5];

determining if a number of neighbor objects [number of times an object (*Element*) is reported] within the expanded radius corresponds to a target number [ $k$ ]; and

continuing to expand the query radius and locate neighbor objects within the expanded radius [See sections 4.2-4.5] until the number of neighbor objects equals or exceeds the target number.

Referring to claims 8-10, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 4.2-4.5 for the details of this disclosure. Hjaltason teaches the method of claim 7, as above, further comprising: calculating a distance of each neighbor object from the query object [See step 18 of Fig. 5] wherein the distance is calculated from the center of the query object [query point] and the distance is based upon boundary-to-boundary distance [distance from  $q$  to bounding box of object (See section 4.4)] as claimed.

Claim 11 is rejected on the same basis as claim 1, in light of the basis for claim 7 above. See the discussion regarding steps 6 – 7 of claim 1 above in light of the basis for claim 7 for the details of this disclosure.

Referring to claim 12, Hjaltason discloses the method for locating neighbor objects as claimed. See section 4.5 for the details of this disclosure. Hjaltason's minimum number of expansions is three in order to cover a full quadrant of nodes in the quad-tree.

Referring to claim 13, Hjaltason discloses the method for locating neighbor objects as claimed. See section 4.8 for the details of this disclosure. Hjaltason teaches the method of claim 11, as above, further comprising:

determining whether an increased number of tiles exceeds a memory capacity in which the database is stored [if number of nodes causes initial build of queue to exceed memory capacity] prior to increasing the number of tiles; and

if the increased number of tiles exceeds the memory capacity then redefining the tiles with an increased size [rebuilding spatial index with larger node regions (See top page 288)].

Referring to claim 14, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 4.2-4.5 for the details of this disclosure. Hjaltason's method further comprises calculating a distance of each located neighbor object from the query object [See line 12 of Fig. 3] if the number of neighbor objects in the initial query radius exceeds the target number to determine the target number of nearest neighbor objects as claimed.

Referring to claims 15 and 16, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 2, 3 and 4.1 for the details of this disclosure. Hjaltason's tiles are rectangular, and all tiles have the same shape and size as claimed.



Referring to claim 17, Hjaltason discloses the method for locating neighbor objects as claimed. See section 2 for the details of this disclosure. Hjaltason's database [GIS database] comprises locations in a geographical region...as claimed.

Referring to claim 18, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 2 and 4.1 for the details of this disclosure. Hjaltason's database is indexed [spatial index] with a linear quadtree as claimed.

Referring to claim 19, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 2-4 for the details of this disclosure. Hjaltason's method further comprises assigning each tile [node] a tile code [key] and an associated geometry [See above] as the tiles are defined as claimed.

Referring to claims 20-22, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 4.2-4.5 for the details of this disclosure. Hjaltason's query radius is expanded by one node each iteration, thus behaving as claimed.

Referring to claims 23 and 24, Hjaltason discloses the method for locating neighbor objects as claimed. See section 4 for the details of this disclosure. Hjaltason's query object is grouped completely or partially within the tiles [nodes] of the search area as claimed.

Claim 25 is rejected on the same basis as claim 17. See the discussion regarding claim 17 above for the details of this disclosure.

Referring to claim 26, Hjaltason discloses the method for locating neighbor objects as claimed. See section 4 for the details of this disclosure. Hjaltason's method

further comprises organizing the neighbor objects in order of distance [reported in order of increasing distance from the query point] as claimed.

Claim 27 is rejected on the same basis as claims 8 and 14. See the discussions regarding claims 8 and 14 above for the details of this disclosure.

Referring to claims 28 and 29, Hjaltason discloses the method for locating neighbor objects as claimed. See sections 4.1-4.6 for the details of this disclosure. Hjaltason's query object comprises a geometry that at least partially intersects at least one tile and the neighbor objects comprise geometries that at least partially intersect at least one tile, [See e.g. section 4.6] wherein at least one tile that the neighbor objects intersect coincides with at least one tile that the query geometry intersects as claimed.

Claims 30 and 31 are rejected on the same basis as claim 7. See the discussion regarding claim 7 as well as the corresponding portions of Hjaltason's article for the details of this disclosure.

### ***Conclusion***

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent No. 6,021,406 to Kuznetsov and U.S. Patent Application Publication No. 2002/0151315 to Hendrey are both considered particularly pertinent to applicant's claimed invention.


The remaining prior art of record is considered pertinent to applicant's disclosure, and/or portions of applicant's claimed invention.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Brian Goddard whose telephone number is 703-305-7821. The examiner can normally be reached on M-F, 9 AM - 5 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Safet Metjahic can be reached on 703-308-1436. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

bdg  
08 January 2004

  
SAFET METJAHIC  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100